07

Validando el Token

Transcripción

[00:00] Listo. Ya estamos en la parte de validar si nuestro toquen aún no está expirado y si está asignado al usuario que ha iniciado sesión. Entonces vamos por partes para esta implementación. Lo primero que tenemos que hacer es venir a nuestro tokenService. Y si yo ya tengo un método para generar mi token ahora yo debería tener un método para digamos consumir el token y validarlo.

[00:27] Voy a creer aquí un método llamado public. Este va a devolver un String getSubject. Entonces el subject, recuerden, es a quién este token está siendo asignado, para quién este token ha sido generado y con eso lo que yo voy a hacer es obtener este nombre de usuario y validar si en efecto el usuario ha iniciado o no ha iniciado sesión en mi sistema.

[00:54] Recuerden que esta es un API, por lo tanto nuestra autenticación fue en stateless y no stateful, pero vamos a ver eso en un momento más. Primero vamos a obtener el subject. ¿Cómo lo hago? Bueno de la misma forma que hicimos con el método para crear nuestro JWT, vamos a inspirarnos en este ejemplo para verificar un JSON Web Token.

[01:17] Entonces lo que vamos a hacer es copiar esta parte del código. No vamos a copiar esto porque el token lo vamos a recibir como parámetro y lo voy a especificar aquí, string token, y pego el código. Ahora lo que tengo que hacer es hacer algunas actualizaciones como por ejemplo, enviarle el tipo de algoritmo correcto que es este de aquí.

[01:45] Y también mi apiSecret, y recuerden, tiene que ser el mismo porque si no es el mismo simplemente esta firma de token no va a coincidir y mi proceso de verificación va a ser rechazado. ¿Cómo funciona? Vamos a borrar esto solo un momento. El issue también tengo que verificarlo por "voll med" y aquí al último le voy a dar un .verify, y lo que va a verificar es mi token. Y con esto ya sería todo.

[02:22] Me está dando un error de compilación porque esto necesita, vamos a ignorar esto, necesita un tipo de verificación que yo no había importado aquí, pero no se preocupen. Vamos a importar también esta excepción y listo, vamos a usar esta que ya está generada y ya está. Tenemos todo en todo en orden para verificar nuestro token.

[02:57] ¿Qué más necesitamos? Aquí lo que yo voy a hacer es hacer un return, porque lo que tengo que hacer ahora es obtener el subject, entonces aquí voy a ponerle punto, getSubject y este verifier yo lo voy a return, verifyer.getSubject(), ahora sí. Perfecto.

[03:41] Vieron, lo que hice fue llevar verifier a una parte afuera del try catch, de forma que una vez que yo lo verifico, verifico el token y voy a retornar el subject de mi proceso de verificación. Vamos a aplicarle una validación, que podría ser tranquilamente esta de aquí o también podemos hacerlo con if, que sería mucho mejor en este caso, porque no estamos escribiendo una prueba unitaria.

[04:11] Entonces podemos decirle que if verifier.getSubject = null. ¿Qué vamos a hacer? Vamos a hacer un throw new RuntimeException, con el mensaje "Verifier invalido". Y si no bueno, retornamos verifier.getSubject, y listo, ya tenemos nuestro método getSubject.

[04:39] Ahora vámonos a nuestro securityFilter porque tenemos que llamarlo ahora, esto está en el tokenService. Yo lo quiero en securityFilter. ¿Qué es lo que tengo que hacer? Exacto, vamos a comenzar declarándolo. Entonces

venimos aquí a SecurityFilter, vengo por acá y voy a decirle private TokenServide con @Autowired, como ustedes ya conocen.

[05:04] Recuerden. Siempre es mejor a nivel de un constructor, en este caso lo estamos haciendo por temas didácticos y la rapidez a nivel del campo, del field, pero no es la idea y aquí es donde vamos a preguntarle por el subject, porque lo que tenemos que hacer es verificar que en efecto ese subject es un usuario que está logueado en mi sistema.

[05:31] Primero que todo voy a hacerle un System.out, para imprimir mi TokenService.getSubject enviándole como parámetro el token y esto debería pues devolverme el subject de mi token. Entonces voy a limpiar todo esto para ver si es que funciona. Esperamos a que cargue.

[05:56] Vemos que ya inició y vamos a tarde enviar. Mi request nuevamente 200 Okay, nada en especial. Venimos a la consola para ver y observen, en efecto, ahora estoy recibiendo mi token que lo estoy imprimiendo aquí y ya tengo al subject del token. ¿Qué quiere decir esto? Ya estoy pasando por el proceso de consumir mi librería de Auth0 para descifrar el token, primero que todo validando la firma, validando la firma del token.

[06:34] Porque si falla en ese nivel me va a lanzar una excepción. Entonces nuevamente tenemos que lograr este exception. Exception.toString, para enterarnos si pasó algo. Va a validar que el issuer, que el emisor de este token sea “voll med” si no también nos va a dar un error.

[06:55] Va a hacer un build de este objeto verifier, y va a verificar nuestro token. Finalmente, ¿qué hacemos? Le damos un verifier.getSubject y con esto vamos a retornar, vamos a validar, si en efecto el subject es válido o no. Ahora, ¿cuál es la pregunta en este momento? Tengo mi SecurityFilter y aquí lo que yo quiero saber es ¿este usuario tiene sesión?

[07:26] Porque como tengo una autenticación del tipo stateless, yo no tengo nada que medir en la memoria, si este usuario ya está logueado o no. Eso ya lo

vamos a implementar en el siguiente video. Nos vemos.